

Pembuatan Kamus Elektronik Kalimat Bahasa Indonesia dan Bahasa Jawa untuk Aplikasi Mobile Menggunakan Interpolation Search

Nur Afifah, Tri Budi Santoso, Mike Yuliana
Laboratorium Sinyal, Politeknik Elektronika Negeri Surabaya
Kampus PENS-ITS, Keputih, Sukolilo, Surabaya.
Email: phiephah@gmail.com, tribudi@eepis-its.edu, mieke@eepis-its.edu

Abstrak

Komunikasi antar manusia memang harus dilakukan karena manusia membutuhkan interaksi dengan manusia lain sebagai makhluk sosial. Proses komunikasi ini pada kondisi tertentu tidak bisa dilakukan karena kendala bahasa. Selain itu, bahasa merupakan cermin budaya dan identitas diri penuturnya, sehingga bahasa harus dilestarikan, misalnya Bahasa Jawa. Masalah tersebut dapat diselesaikan jika terdapat orang ketiga yang menguasai kedua bahasa. Orang ketiga ini disebut sebagai penerjemah.

Pada paper ini telah dibuat *software* kamus elektronik atau translator *J2J* untuk menerjemahkan kalimat dari bahasa Indonesia ke bahasa Jawa atau sebaliknya. Pembuatan *software* ini menggunakan pemrograman *J2ME*. Proses penerjemahan kalimat dimulai dari pemecahan kalimat dengan pembacaan tanda spasi, pencarian padanan kata dengan metode *interpolation search*, dan penggabungan kata menjadi kalimat kembali.

Hasil penelitian yang diperoleh dari pengujian penerjemahan kalimat bahasa Indonesia ke bahasa Jawa untuk jumlah kata sama dengan 5, diperlukan rata-rata waktu 12.24 milisecond pada *emulator*, dan 52.8 milisecond pada ponsel, sedangkan dari bahasa Jawa ke Indonesia diperlukan rata-rata waktu 11.68 milisecond pada *emulator*, dan 38.48 milisecond pada ponsel. Selain itu dari kuisioner didapat data bahwa 100% *responden* mengatakan bahwa aplikasi *J2J* ini membantu dalam menerjemahkan bahasa Jawa dimana aplikasi ini bisa dijalankan dimanapun, dan kapanpun dengan media perangkat *mobile* yang praktis.

Kata Kunci: Kamus Elektronik, *J2ME*, Interpolation Search

1. Pendahuluan

Komunikasi antar manusia memang harus dilakukan karena manusia membutuhkan interaksi dengan manusia lain sebagai makhluk sosial. Proses komunikasi ini pada kondisi tertentu tidak bisa dilakukan karena kendala bahasa. Komunikasi antara 2 orang hanya bisa dilakukan dengan sebuah bahasa jika keduanya memahami bahasa tersebut.

Kenyataan di dunia nyata tidak selalu demikian karena ada kemungkinan seseorang menguasai sebuah bahasa sedangkan yang lain tidak menguasai dan hanya menguasai bahasa yang lain. Dalam kondisi seperti ini komunikasi tidak mungkin dilakukan. Selain itu, bahasa merupakan cermin budaya dan identitas diri penuturnya, sehingga bahasa harus dilestarikan. Karena hilangnya keanekaragaman bahasa di Indonesia, berarti hilangnya budaya bangsa yang merupakan kerugian tak terhitung nilainya, misalnya Bahasa Jawa.

Masalah tersebut dapat diselesaikan jika terdapat orang ketiga yang menguasai kedua bahasa. Karena menguasai kedua bahasa, orang ketiga ini dapat berkomunikasi dengan kedua orang yang berbeda bahasa tersebut. Dengan kemampuan ini, orang ketiga tersebut bisa menjadi media penyambung komunikasi antara kedua orang tersebut, yang sebelumnya tidak bisa dilakukan. Orang ketiga ini disebut sebagai penerjemah.

Penerjemah ini sangat diperlukan dalam era informasi global ini. Banyak informasi yang disampaikan orang lain dalam bahasa yang berbeda, baik dalam bentuk lisan maupun tulisan. Sehingga sangat diperlukan sebuah *software* (perangkat lunak) yang dapat menerjemahkan dari sebuah bahasa ke bahasa yang lain.

Pada proyek akhir ini dibuat sebuah perangkat lunak yang dapat menerjemahkan kalimat dari Bahasa Indonesia ke Bahasa Jawa dan sebaliknya atau yang biasa disebut *translator*. *Software* ini akan dipasang pada perangkat komunikasi *mobile* (telepon genggam), sehingga orang bisa mengakses *software* ini dengan mudah dan praktis dimanapun dan kapanpun. *Software* ini bernama *J2J* (Java to Java) *mobile dictionary*.

2. Teori Penunjang

2.1 Kamus Elektronik

Dalam penelitian ini, yang dimaksudkan sebagai kamus elektronik adalah kamus yang berupa piranti lunak dan bisa diinstal ke komputer[1]. Dalam hal ini, kamus elektronik yang tidak bisa diinstal ke komputer, misalnya kamus yang dipasarkan oleh *AlfaLink* tidak termasuk

dalam pembahasan. Alasan utamanya adalah karena kamus tersebut menggunakan alat tertentu yang harganya cukup mahal dan tidak bisa diinstal ke komputer, sehingga penyebarannya sangat terbatas. Terdapat dua versi kamus elektronik *bilingual* Inggris-Indonesia yang cukup luas tersedia, yaitu *Linguist Version 1.0* dan *Indict Version 2.0*.

2.2 J2ME Wireless Toolkit

J2ME *Wireless Toolkit* adalah sekumpulan *tool* yang digunakan untuk mengembangkan aplikasi-aplikasi dalam handphone dan *mobile device* lainnya. Sun Microsystems telah menyediakan J2ME *Wireless Toolkit* (J2ME WTK) untuk mengembangkan aplikasi-aplikasi dalam *handphone*. J2ME WTK berbasiskan pada CLDC dan MIDP. J2ME WTK adalah program yang meniru kerja ponsel yang mendukung MIDP atau yang biasa disebut *emulator*. Oleh karena itu, belum tentu MIDlet yang berjalan di *emulator* juga berjalan pada ponsel yang sebenarnya, karena juga bergantung pada kemampuan dan kapasitas ponsel yang digunakan.

Pada *Netbeans 6.8* terdapat emulator bawaan, yaitu *Java ME Platform SDK 3.0*, yang memiliki jenis emulator lebih banyak dibandingkan dengan J2ME WTK.

2.3 Interpolation Search

Interpolation Search adalah algoritma pencarian yang lebih efisien daripada algoritma *Binary* dan *Sequential Search*. Hal ini dikarenakan algoritma ini tidak perlu menjelajahi setiap elemen dari tabel. Kerugiannya adalah algoritma ini hanya bisa digunakan pada tabel yang elemennya sudah terurut baik menaik maupun menurun.

Sama seperti *Binary*, teknik ini hanya dapat dilakukan pada list yang telah terurut dan berada pada struktur array dan data yang dicari diperkirakan ada di dalam list. Teknik ini menemukan item dengan memperkirakan seberapa jauh kemungkinan item berada dari posisi saat itu dan pencarian berikutnya. Teknik ini juga dilakukan pada list yang sudah terurut.

Rumus umum *Interpolation Search* bisa dilihat pada persamaan (1) :

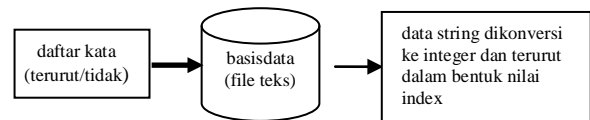
$$P = \frac{\text{kunci} - k[\min]}{k[\max] - k[\min]} \quad (1)$$
$$\text{Posisi} = \text{Round}(P * (\max - \min)) + \min$$

3. Metodologi

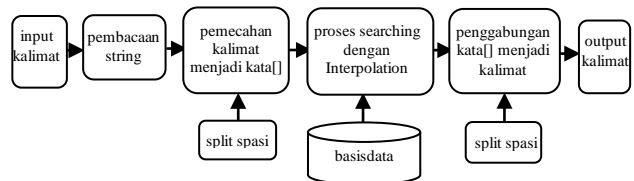
3.1 Perencanaan Sistem

Perancangan sistem didasarkan pada pembuatan basisdata standar yang berisi kumpulan kata-kata yang disimpan pada file teks(.txt). Metode pencarian yang digunakan adalah

Interpolation yang mengharuskan data (yang diolah) sudah dalam kondisi terurut atau yang disebut *leksikon*. Pada pembuatan aplikasi ini tidak menggunakan metode *sorting* untuk mengurutkan kata-kata tersebut, tetapi kata-kata tersebut(string) akan dikonversi menjadi index yang otomatis terurut (integer) untuk selanjutnya dilakukan proses perhitungan pencarian posisi dengan rumus *interpolation*. Sehingga daftar kata (string) pada basisdata tidak harus terurut. Kesemuanya itu akan dihubungkan ke dalam satu *interface* yang dibuat menggunakan *Netbeans*, dan akan dijalankan dengan *emulator*.



Gambar 1. Pembuatan Database Standar



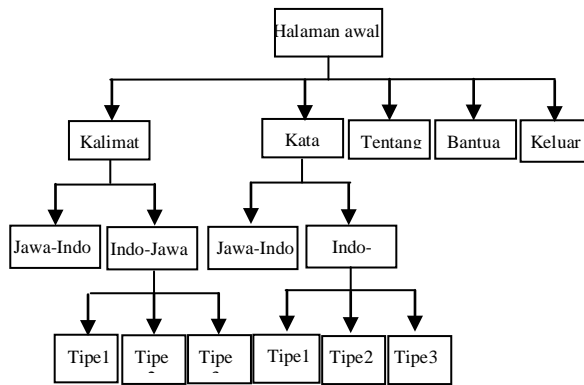
Gambar 2. Proses Penerjemahan Kalimat

3.2 Pembuatan Software

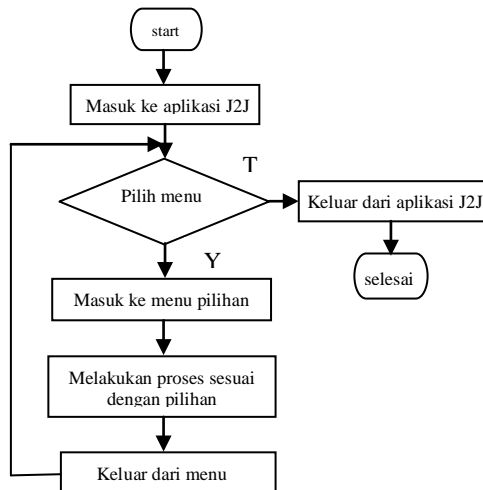
3.2.1 Perancangan Basis Data Standar

Basisdata berfungsi sebagai tempat penyimpanan kata. Karena dalam hal ini dibuat *software translator*, sehingga akan dibutuhkan banyak sekali kata baik dalam bahasa Indonesia dan bahasa Jawa. Basisdata yang dibuat disini tidak menggunakan pemrograman khusus untuk *database*, namun hanya memanfaatkan file teks yang disimpan pada notepad yang akan disimpan dengan ekstensi .txt. Sehingga nantinya akan ada perintah yang mengintegrasikan file teks ke dalam software. Pada kasus ini, basisdata dibagi menjadi dua macam yaitu basisdata yang berisi padanan kata Bahasa Indonesia ke Bahasa Jawa (disimpan dengan nama *DaftarKataIndoJawa.txt*) dan basisdata yang berisi padanan kata Bahasa Jawa ke Bahasa Indonesia (disimpan dengan nama *DaftarKataJawaIndo.txt*). Kata-kata pada basisdata tidak harus terurut, karena akan dibuat program tersendiri untuk merubah kata (string) menjadi index (integer) sebelum proses *interpolation search* dijalankan.

3.2.2 Perancangan Interface



Gambar 3. Alur Desain Sistem



Gambar 4. Flowchart Kinerja Sistem

3.2.3 Proses Pemecahan Kalimat

Pada Proyek Akhir ini akan digunakan Metode Pemecahan Kalimat dengan *Split* untuk kebutuhan memecah atau memisah kalimat menjadi kata-kata dengan cara yang sederhana. Kalimat yang dimasukkan akan di cek jumlah katanya, selanjutnya akan dilakukan proses split. Dimana setiap kata yang bertemu dengan tanda spasi(' '), maka kata tersebut akan disimpan kedalam array. Proses *split* ini terus dieksekusi sampai batas jumlah kata yang dimasukkan.

• Simulasi Pemecahan Kalimat

Saya berangkat ke kampus \Rightarrow kalimat

Saya berangkat ke kampus
 ↑ ↑ ↑
 spasi spasi spasi

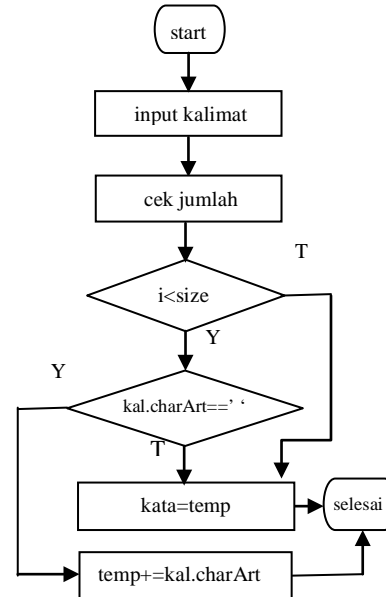
Saya	Berangkat	Ke	Kampus
Kata[1]	Kata[2]	Kata[3]	Kata[4]

• Algoritma Pemecahan Kalimat

1. Input kalimat

2. Cek jumlah kata atau panjang kata
3. Split, `kal.charArt(kata)` bertemu dengan spasi ' '
4. Bandingkan `i` kurang dari `size`
5. Jika lebih kecil, proses split, dan `temp+=kal.charArt`
6. Jika lebih besar, `temp=kata`, selesai.

• Flowchart Pemecahan Kalimat



Gambar 5. Flowchart Pemecahan Kalimat

3.2.4 Proses Searching

• Simulasi Interpolation Search

Cara kerja metode pencarian interpolasi dapat disimulasikan sebagai berikut, dimisalkan kita memiliki data terurut seperti di bawah ini:

Contoh :

index	konversi	Jawa	Indonesia
[0]	0	Aku	Saya
[1]	1	Bocah	Anak
[2]	2	Budhal	Berangkat
[3]	3	Dolan	Main
[4]	4	Getih	Darah
[5]	5	Kali	Sungai
[6]	6	Kanca	Teman
[7]	7	Kodanan	Kehuianan

Kunci pencarian (konversi)? 3

Posisi $[index] = (3-0)/(7-0)*(7-0)+0=3$

Kode [3]==kunci? Ya \Rightarrow Dolan

\Rightarrow Main

Kunci pencarian (konversi)? 9

Posisi $[index] = (3-0)/(7-0)*(7-0)+0=9$

Kode [9]==kunci?

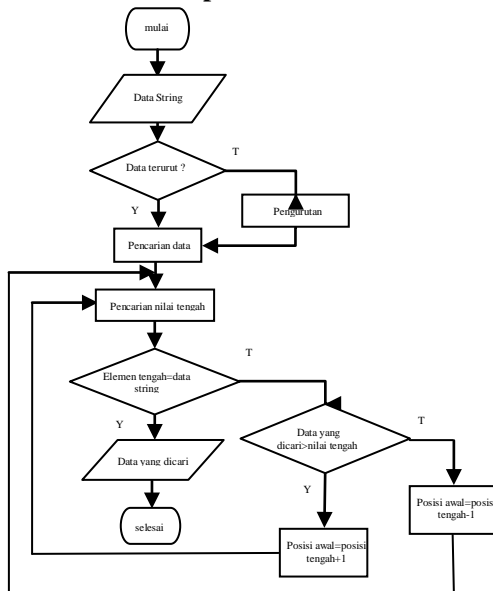
- tidak \Rightarrow kode tidak ada dalam data

• Algoritma Interpolation Search

1. Banyaknya record array (k)
2. Nilai awal `min=0` ; `max=k-1`

3. Hitung $mid = min + ((kunci - k[min]) * (max - min)) / (k[max] - k[min])$
4. Bandingkan data yang dicari(kunci) dengan data posisi tengah(mid)
5. Jika lebih kecil, proses dilanjutkan dengan posisi $max = posisi\ tengah - 1$
6. Jika lebih besar, proses dilanjutkan dengan posisi $min = posisi\ tengah + 1$
7. Jika data posisi tengah(mid) = data yang dicari(kunci), maka $index = mid$, selesi
8. Jika $min \leq max$ dan $k[mid] \neq kunci$, maka ulangi langkah 3
9. Jika $k[mid] \neq kunci$, maka $index = -1$, selesi.

• Flowchart Interpolation Search



Gambar 6. Flowchart Interpolation Search

3.2.5 Proses Penggabungan Kata

• Simulasi Penggabungan Kata

Berikut merupakan hasil terjemahan kata saat proses pencarian.

Saya	Berangkat	Ke	Kampus
Kata[1]	Kata[2]	Kata[3]	Kata[4]
Aku	Budhal	Nyang	Kampus

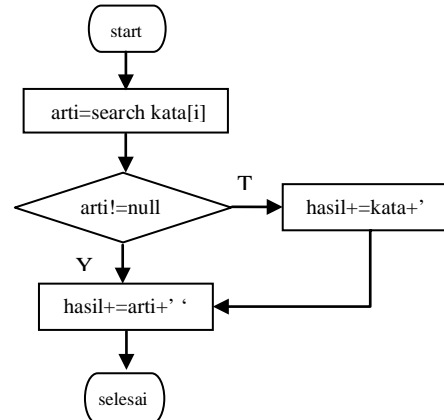
Hasil dari terjemahan tiap kata akan dimasukkan ke array sesuai dengan index masing-masing. Setelah proses pencarian dan pengisian array selesai, maka kata-kata tersebut akan digabung kembali menjadi sebuah kalimat dengan pemisahan tanda spasi.

Aku budhal marang kampus
 ↑ ↑ ↑
 spasi spasi spasi
 Aku budhal marang kampus → kalimat

• Algoritma Penggabungan Kata

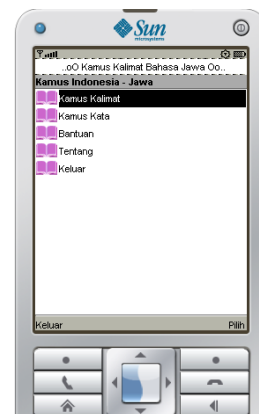
1. Banyaknya arti kata yang sudah di-searching
2. Bandingkan apakah arti \neq null
3. Jika ya, $hasil += arti + ' '$
4. Jika tidak, $hasil += kata[i] + ' '$
5. Selesi

• Flowchart Penggabungan Kata



Gambar 7. Flowchart Penggabungan Kata

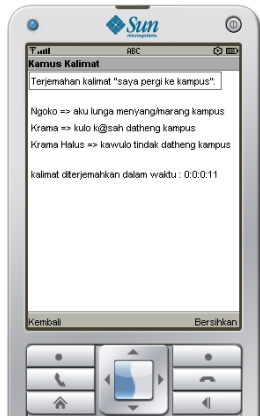
4. Hasil dan Analisa



Gambar 8. Tampilan Menu



Gambar 9. Tampilan Kamus Kalimat



Gambar 10. Tampilan Hasil Terjemahan

4.1 Pengujian Pemecahan Kalimat

Proses awal sebelum kalimat diterjemahkan adalah proses split/pemecahan kalimat menjadi kata-kata, untuk kemudian di artikan satu-persatu

Berikut adalah hasil compiler ketika menerjemahkan sebuah kalimat “aku mangan sego” :

```
Running in the manufacturer security domain
waktu untuk proses split kalimat : 0:0:0.0 milliseconds
kata yang dicari berada pada index ke- 1111
kamu ----- kowe=kamu
waktu pencarian Java-Indo: 0:0:0.2
waktu untuk proses penggabungan : 0:0:0.0 milliseconds
kata yang dicari berada pada index ke- 1208
makan ----- mangan=makan
waktu pencarian Java-Indo: 0:0:0.1
waktu untuk proses penggabungan : 0:0:0.0 milliseconds
kata yang dicari berada pada index ke- 1347
nasi ----- sego=nasi
waktu pencarian Java-Indo: 0:0:0.2
waktu untuk proses penggabungan : 0:0:0.0 milliseconds
```

Gambar 11. Hasil Compiler Pemecahan Kalimat

4.2 Pengujian Kalimat Indonesia-Jawa

Pengujian untuk pencarian terjemahan kata atau kalimat dilakukan pada emulator J2ME WTK 2.5.2 dan pada perangkat *mobile*, yaitu ponsel tipe NOKIA 6600.

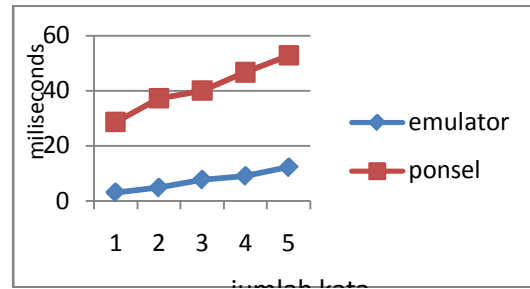
Dilakukan 5 kali percobaan dengan kalimat yang berdeda, dan masing-masing kalimat tersebut 10 kali percobaan dengan jumlah data yang sama yaitu sekitar 1000 data.

Tabel 1. Perbandingan Durasi Indonesia-Jawa

Jumlah Kata	Waktu emulator(ms)	Waktu ponsel(ms)
1	3.1	28.68
2	4.82	37.22
3	7.72	40.08
4	9.08	46.66
5	12.24	52.8

Dari tabel di atas diperoleh grafik di bawah ini, terlihat bahwa rata-rata waktu yang diperlukan akan semakin banyak, jika jumlah kata yang diterjemahkan juga semakin banyak. Waktu yang diperlukan emulator untuk proses

penerjemahan lebih cepat dibandingkan dengan perangkat *mobile* atau ponsel.



Gambar 12. Grafik perbandingan Durasi untuk Indonesia-Jawa

4.3 Pengujian Kalimat Jawa-Indonesia

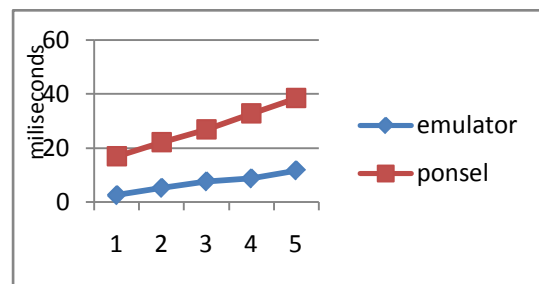
Jumlah data pada untuk database kata Jawa-Indonesia yaitu sekitar 2500 data.

Tabel berikut merupakan hasil waktu rata-rata yang diperoleh dari tabel-tabel sebelumnya dalam pengujian untuk menerjemahkan kalimat Jawa ke Indonesia.

Tabel 2. Perbandingan Durasi Jawa-Indonesia

Jumlah Kata	Waktu emulator(ms)	Waktu ponsel(ms)
1	2.52	16.86
2	5.2	21.92
3	7.6	26.94
4	8.7	32.88
5	11.68	38.48

Dari tabel di atas diperoleh grafik di bawah ini, terlihat bahwa rata-rata waktu yang diperlukan akan semakin banyak, jika jumlah kata yang diterjemahkan juga semakin banyak. Waktu yang diperlukan emulator untuk proses penerjemahan lebih cepat dibandingkan dengan perangkat *mobile* atau ponsel.



Gambar 12. Grafik perbandingan Durasi untuk Jawa-Indonesia

4.4 Pengujian Penggabungan Kalimat

Proses penggabungan dilakukan pada setiap kata yang telah ditemukan/diterjemahkan. Sehingga satu demi satu akan digabungkan.

Berikut adalah hasil compiler ketika menggabungkan sebuah kalimat:

```
Running in the manufacturer security domain
waktu untuk proses split kalimat : 0:0:0:0 milliseconds
kata yang dicari berada pada index ke- 1111
kamu
waktu pencarian Java-Indo: 0:0:0:2
waktu untuk proses penggabungan : 0:0:0:0 milliseconds
kata yang dicari berada pada index ke- 1208
makan
waktu pencarian Java-Indo: 0:0:0:1
waktu untuk proses penggabungan : 0:0:0:0 milliseconds
kata yang dicari berada pada index ke- 1347
nasi
waktu pencarian Java-Indo: 0:0:0:2
waktu untuk proses penggabungan : 0:0:0:0 milliseconds
```

penggabungan ke-1
kamu

penggabungan ke-2
kamu+makan

penggabungan ke-3
kamu+makan+nasi

Gambar 13. Hasil Compiler Penggabungan Kata

4.4 Pengujian Basisdata

Pengujian basisdata ini dimaksudkan untuk menguji tanda pemisah yang digunakan pada *file teks*, dimana pada program yang telah dibuat telah digunakan tanda pemisah “#” untuk memisahkan antar kata satu dengan kata yang lain. Selain tanda tersebut juga bisa digunakan tanda pemisah lain. Karena pada J2ME tidak terdapat fungsi untuk membaca ganti baris atau enter.

Tabel berikut adalah data pengujian tanda pemisah pada basisdata yang digunakan:

Tabel 6. Pengujian Tanda Pemisah

Tanda Pemisah	Kondisi
#	Berhasil
!	Berhasil
\$	Berhasil
%	Berhasil
^	Berhasil
*	Berhasil
(Berhasil
)	Berhasil
+	Berhasil
<	Berhasil
>	Berhasil
?	Berhasil
:	Berhasil
;	Berhasil
“	Berhasil
‘	Berhasil
Spasi	Gagal
@	Gagal
/	Gagal
-	Gagal
Huruf	Gagal
=	Gagal

Pada tabel di atas terlihat bahwa terdapat 2 hasil kondisi untuk pengujian tanda pemisah, yaitu berhasil dan gagal. Dimana untuk kondisi berhasil yaitu kalimat atau kata berhasil diterjemahkan dengan baik dan benar. Sedangkan untuk kondisi gagal yaitu kalimat atau kata tidak bisa diterjemahkan dengan sempurna. Kata yang diterjemahkan terkadang tidak sesuai dengan kata yang dicari, karena tanda pemisah yang digunakan rancu/ambigu dengan tanda yang juga telah digunakan untuk fungsi yang lain.

4.5 Pengujian dari Hasil Kuisisioner

Pengujian lain dilakukan dengan menyebarkan kuisisioner mengenai aplikasi *J2J* ini ke 30 *responden* secara acak.

Tabel 7. Hasil Kuisisioner

No	Poling	Jumlah	Prosentase
1.	Lebih menyukai kamus elektronik	29	96.67%
2.	Lebih menyukai kamus manual	1	3.33%
3.	Yang memahami bahasa Jawa	18	60%
4.	Tidak memahami bahasa Jawa	12	40%
5.	Sangat menarik aplikasi <i>J2J</i> ini	21	70%
6.	Cukup menarik aplikasi <i>J2J</i> ini	9	30%
7.	Kurang menarik aplikasi <i>J2J</i> ini	0	0%
8.	Pengoperasian aplikasi <i>J2J</i> mudah	28	93.33%
9.	Pengoperasian aplikasi <i>J2J</i> sedang	2	6.67%
10.	Pengoperasian aplikasi <i>J2J</i> sulit	0	0%
11.	Terbantu dengan adanya aplikasi <i>J2J</i>	30	100%
12.	Tidak terbantu dengan	0	0%
13.	Pernah menggunakan aplikasi sejenis ini sebelumnya	5	16.67%
14.	Belum pernah menggunakan aplikasi sejenis ini sebelumnya	25	83.33%

5. Kesimpulan

Dari hasil pengujian dan analisa, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Pada kamus elektronik yang telah diuji diperoleh hasil sesuai yang diharapkan yaitu *software* yang dapat menerjemahkan kalimat bahasa Indonesia ke dalam bahasa Jawa ataupun sebaliknya.
2. Pada pengujian kalimat dari bahasa Indonesia ke Jawa untuk kata yang berjumlah 5, diperlukan rata-rata waktu 12.24 milisecond pada *emulator*, dan 52.8 milisecond pada *ponsel*.
3. Pada pengujian kalimat dari bahasa Jawa ke Indonesia untuk kata yang berjumlah 5, diperlukan rata-rata waktu 11.68 milisecond

pada *emulator*, dan 38.48 milisecond pada *ponsel*.

4. Pencarian data menggunakan metode *Interpolation Search* secara teori yaitu menghitung dan menentukan posisi index, maka diperoleh hasil dengan index yang sama antara teori dengan pengujian secara praktik menggunakan program yang telah dibuat.
5. Pengujian basisdata dilakukan untuk menguji tanda pemisah antar kata pada file teks, dan beberapa karakter yang tidak bisa digunakan adalah tanda spasi, “-“, “@”, “=”, “/” dan huruf, tanda tersebut menimbulkan kerancuan ketika dilakukan proses pencarian kata ke file teks karena tanda tersebut telah digunakan untuk fungsi yang lain.
6. Kuisisioner yang didapat dari 30 *responden* menunjukkan bahwa aplikasi *J2J* ini secara praktis membantu masyarakat dalam menerjemahkan bahasa Jawa dan bahasa Indonesia dengan prosentasi jumlah yang memilih yaitu 100%, dan 96.67% *responden* lebih menyukai kamus elektronik dibanding kamus manual.

6. Daftar Pustaka

- [1] Bambang Hariyanto, Ir., MT.. *Esensi-esensi bahasa pemrograman Java*. Bandung: Informatika, 2007.
- [2] Budi Raharjo, Imam Heryanto, Arif Hatyono. *Tuntunan Pemrograman Java untuk Handphone*. Bandung: Informatika, 2008.
- [3] Dewi Martina Andayani, *Tugas Akhir Pembuatan Kamus Elektronik Kata-Kata Bahasa Indonesia-Jawa Menggunakan Metode Binary Search Berbasis Perangkat Lunak*, PENS-ITS, Surabaya, 2009.
- [4] Jurianto, Drs., *Pengembangan Kamus Elektronik Akuntansi berbasis Korpus*, Universitas Airlangga, 2007.
- [5] M. Shalahuddin, Rosa A.S.. *Pemrograman J2ME*. Bandung: Informatika, 2008.
- [6] Savitch, Walter. *Java An Introduction to Computer Science and Programming*. USA: Pearson Education Inc. 2004.
- [7] Setiawan, Sandi. *Artificial Intelligence*. Yogyakarta: Andi Offset, 1993.
- [8] <http://forum.nokia.com>.
- [9] <http://java.sun.com/j2me/>
- [10] <http://www.java2s.com/Code/Java/J2ME/>
- [11] <http://www.wikipedia.org/Interpolation/>